Perception Project

Liyuan Liu

October 11, 2018

1 Introduction

In this project, a 3D point cloud process pipeline is carried out. Since the point cloud data has much noise and only a small portion of the data is what we need, the first step is going through a series of filters. After filtering, the data is much smaller than what it used to be. Next, it comes how to segment objects from one to the other. The second step is to segment the objects. In this project, the number of objects is not known before hand. It is a good idea to use **DBSCAN** clustering algorithm. Finally, we need to extract features from the segmented objects and identify what it is. The third step is to classify the segmented objects into different groups.

2 Filtering

The raw 3D point cloud data looks like below,

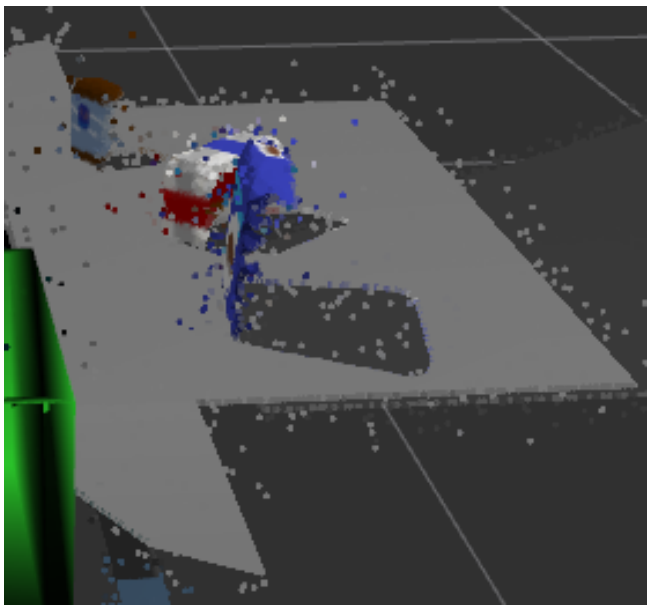

Figure 1: Raw data of 3D point cloud

In the filtering part, we use 4 filters. They are voxel filter, passthrough filter, ransac filter and statistical filter. Since the point cloud data is very dense, handling all these data is very time consuming, we need to use voxel filter to reduce the size of data and not lose much information.

After going through voxel filter, the image look like this,

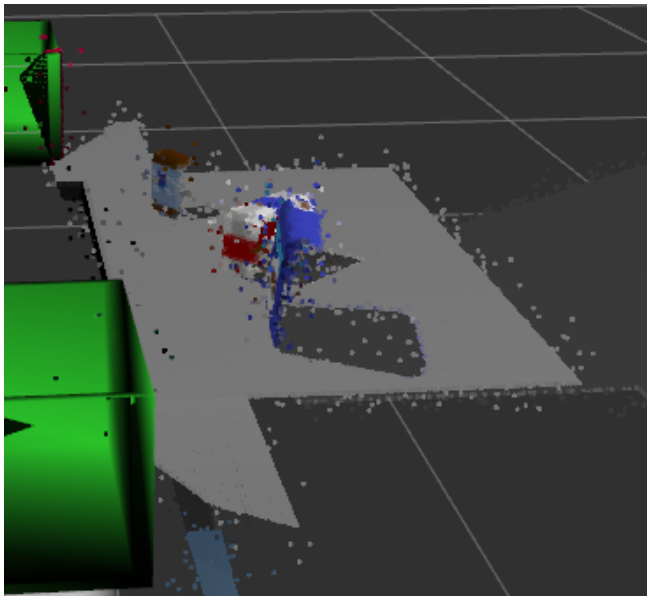

Figure 2: After voxel filter

What we focus on are the objects on the table, thus we want to narrow the scope. The passthrough filter can be used.

After using passthrough filter, we got

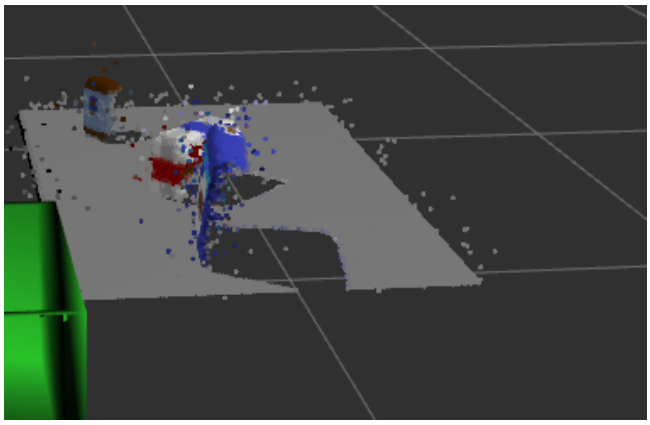

Figure 3: After passthrough filter

To get rid of the influence of table, we can use RANSAC algorithm to filter out the table.

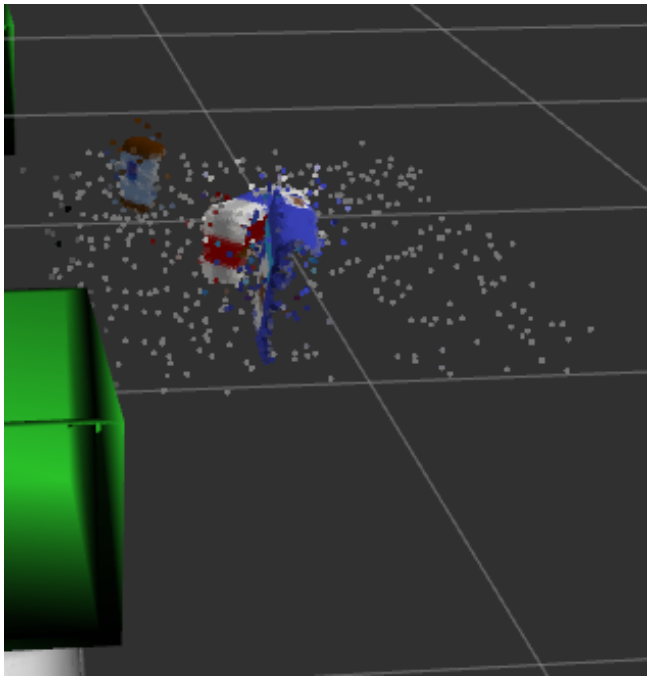

Figure 4: After ransac filter

As can be seen, there are many noisy point around objects. Statistical removal filter is a good tool to make less noise.

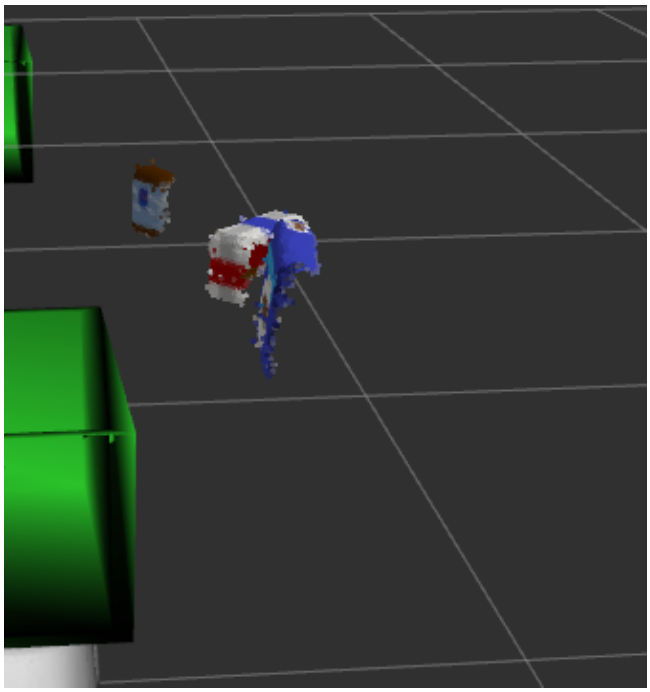

Figure 5: After statistic removal filter

3 Segmentation

After filtering, we want to group points to identify objects. This is a clustering problem. There are many clustering algirithm, in this project we use **DBSCAN**.

After implementing DBSCAN clustering algorithm, we got

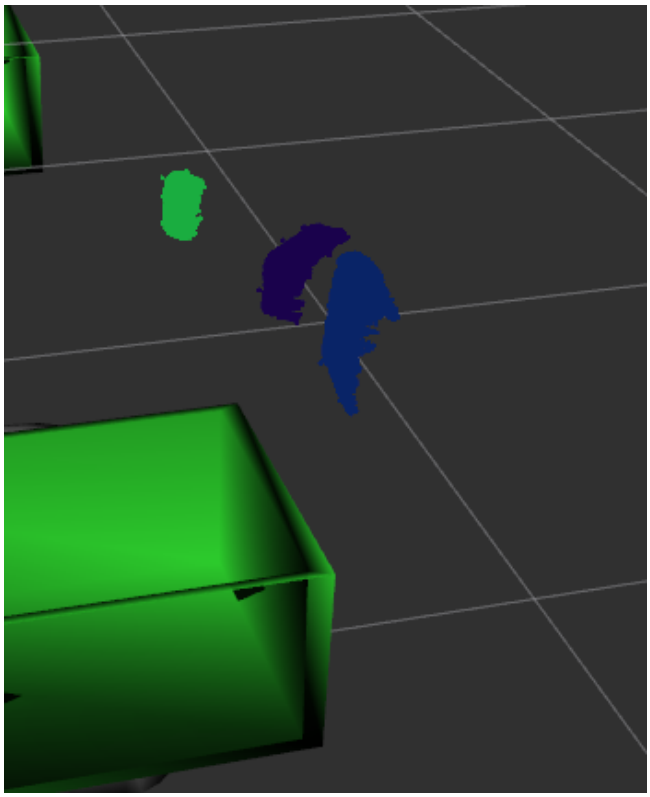

Figure 6: Segmentation

4 Objection Recognition

To identify what the object is, we need to implement some classification algorithm. In this project we use **Support Vector Machine** classifier. This is a manchine leanring algorithm. Thus many data of the objects we want to identify need to be collected.

```
# in one terminal
roslaunch sensor_stick training.launch
# in another terminal
rosrun sensor_stick capture_features_proj3.py
```

Then we can get 200 data for each category.

After we have data, we can train SVM.

```
# in one terminal
rosrun sensor_stick train_svm_proj3.py
```

The confusion matrix,

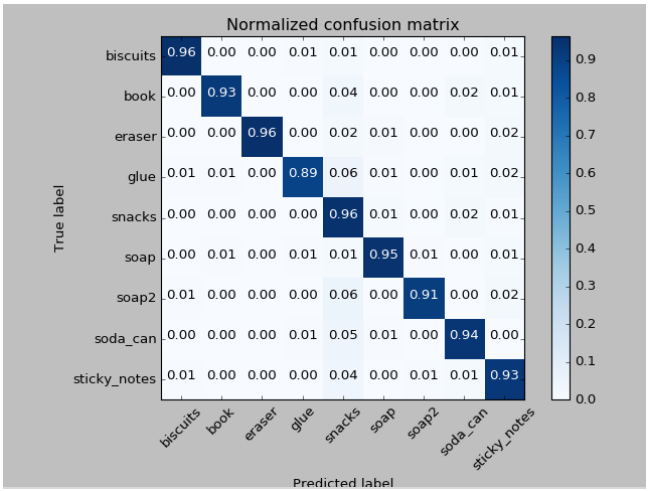

Figure 7: Normalized confusion matrix

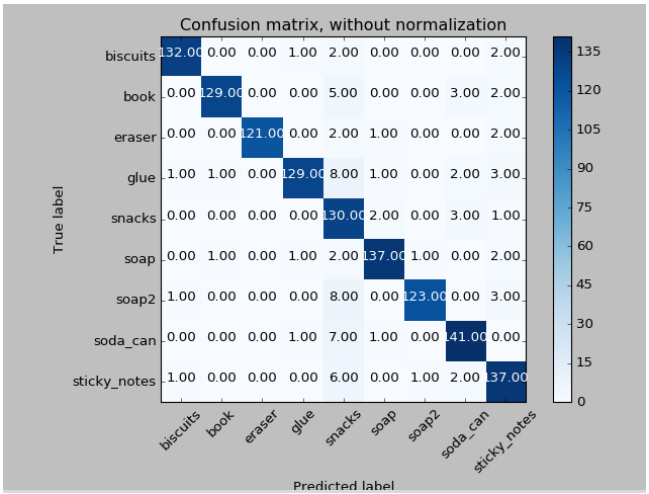

Figure 8: Confusion matrix

The kernel of SVM is **linear** and the value of C is 0.1. The model file name is *model_200.sav*

Then using the model above and the objects segmented, we can classify them.

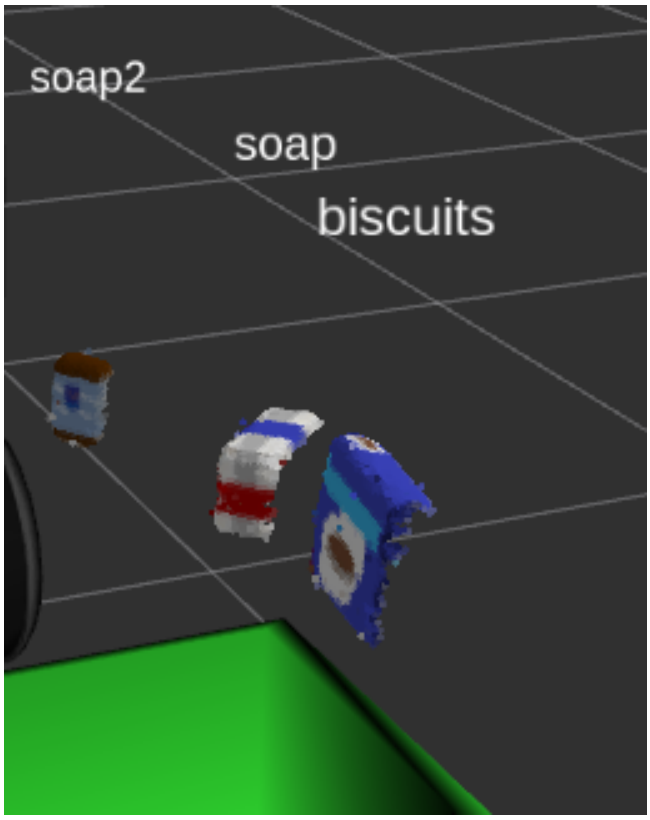

Figure 9: Recognition

5 Pick and place

The classification results for all three worlds are shown below. The accuracy rate of world1 is 100%. The accuracy rate of world2 is 100%. And the accuracy rate of world3 is 87.5%.

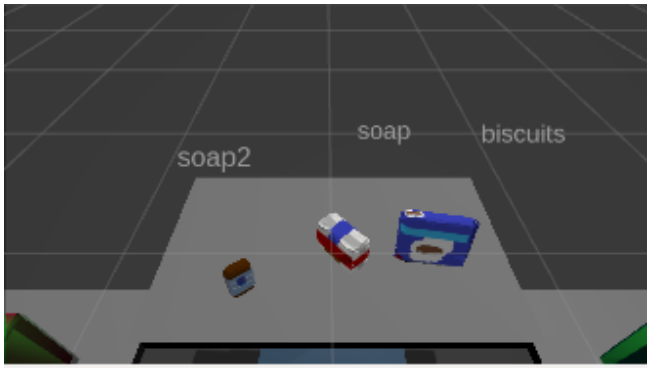

Figure 10: World1

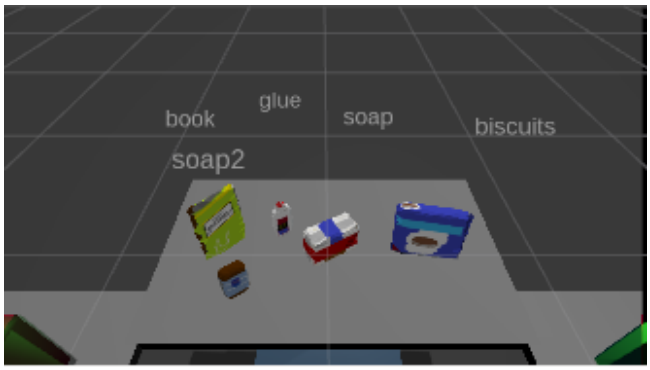

Figure 11: World2

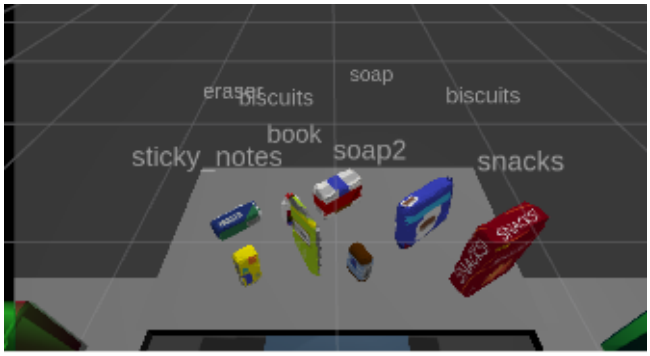

Figure 12: World3